HEFS Hindcasting Guide

Version: OHD-CORE-CHPS-4.4.a

Release Date: 16 May 2016

National Weather Service
Office of Hydrologic Development

Table of Contents

1 Overview

There are three components for which HEFS hindcasts may be generated:

1. **MEFP**: Generate ensembles of precipitation and temperature forcing time series using archived forecasts or reforecasts available for forecast sources employed in the MEFP.
2. **Hydrologic Models**: Generate ensemble of streamflow time series using MEFP hindcast forcing ensembles as input.
3. **EnsPost**: Generate post-processed streamflow ensemble by applying the HEFS EnsPost to the streamflow time series output by the hydrologic models.

The output from MEFP, the hydrologic models, and EnsPost are exported for verification by the Ensemble Verification System (EVS).

This document provides instructions for the following:

- **Configuring a hindcasting SA to generate HEFS hindcasts.**
- **Configuring modules to export results to PI-timeseries XML (or .fi/.bin) files for use in verification with EVS.**
- **Generating hindcasts using the hindcasting SA.**

The Ensemble Verification System (EVS) can be used to perform verification using the exported PI-timeseries file described above. See the *EVS Manual* for more information:

http://www.nws.noaa.gov/oh/hrl/general/HEFS_doc/EVS_MANUAL.pdf

HEFS validation guidance is provided at the following link:

http://www.nws.noaa.gov/oh/hrl/general/HEFS_doc/Validation_Guidance/RFC_validation_guide_FINAL.pdf

Examples of HEFS hindcasting and verification (phased validation) can be found here:

http://www.nws.noaa.gov/oh/hrl/hsmb/hydrologic_ensembles/publications_presentations/index.html

1.1 *Notation*

Within this document, the following notation is used:

- All graphical interface components are **Capitalized and in Bold**.
- All XML snippets are in this font.
- All complete XML example files are in `this multi-colored font` to allow for easier identification of XML elements, attributes, and comments.
- All command line entries are in this font.
- All important terms defined in the Section 1.2, Terminology, are *italicized*.

1.2 *Terminology*

- *hindcasting standalone (SA)*: The standalone used to generate hindcasts for HEFS components, setup in Section 1.4.
- *hindcast workflow*: The workflow to be executed when generating the HEFS hindcast, after the warm states have been created.
- *Hindcast T0*: A forecast time (T0) for which a hindcast is to be generated.

1.3 *Directories of Note*

The following directories will be referred to in the instructions provided below:

- <*region_dir*>: The *hindcasting standalone* (see Section 1.4) region home directory, typically "##rfc_sa".
- <*configuration_dir*>: The standalone Config directory, typically <*region_dir*>/Config.
- <*export_root_dir*>: The directory under which PI-timeseries files will be exported for verification by EVS; typically <*region_dir*>/Export.

1.4 *Pre-installation Steps*

1. Create a CHPS SA to be used to generate hindcasts. The SA should be setup identical to the operational HEFS SA, so that all HEFS components for which hindcasts are to be generated are configured for operational ensemble generation. This is referred to as the *hindcasting SA*.

2 Configuration Changes for Hindcasting

This section provides instructions for the following:

- **Configuring MEFP for hindcasting.**
- **Configuring exports of appropriate time series to PI-timeseries Fastinfoset-binary (.fi/.bin) files.**
- **Modifying other modules as needed for hindcasting.**

2.1 *Configuration File Changes for MEFP Hindcasting (Required)*

Changes to make to enable MEFP hindcasting include the following:

1. Remove (comment-out) MEFP Preprocessing Components.
2. Set the MEFP_HINDCASTING Global Property.
3. Configure all modules that execute the MEFPEnsembleGeneratorModelAdapter for Hindcasting.

Each step is described in the following sections.

2.1.1 Remove (Comment-out) MEFP Preprocessing Components

Action: For the *hindcasting SA*, remove, or comment out, all references in the configuration to the MEFP pre-processing workflows that import, interpolate, and process operational gridded forecasts. Within the default installation of MEFP, that includes references to workflows with these workflowIds:

MEFP_Preprocess_RFC_Forecast
MEFP_Preprocess_GEFS_Forecast
MEFP_Preprocess_CFSv2_Forecast

The names of the workflows that must be commented out and the names of the files that require modification may vary by RFC.

Description: Operational gridded forecasts are not appropriate for use in generating hindcasts. The reforecasts or archived forecasts necessary for MEFP hindcasting are acquired from the parameter file for each location for which MEFP is executed. The pre-processing components of MEFP must, therefore, not be executed or errors will occur.

It may be necessary to comment out references to other pre-processing workflows associated with forecast sources added beyond the default installation of MEFP. For example, see the WPC QPF (see the *WPC QPF MEFP Plugin Configuration Guide*) which may include an operational pre-processing step not appropriate to be called during hindcasting.

See the example below of the default delivered MEFP_Forecast.xml configuration file with the calls to the MEFP pre-processing workflows commented out (note the gray shaded lines).

Again, when hindcasting, the MEFPEnsembleGeneratorModelAdapter acquires all data required as input for the forecast sources in order to generate an ensemble from the parameter (.tgz) file for the location involved.

EXAMPLE: Modified Delivered MEFP_Forecast.xml Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/workflow.xsd" version="1.1">

    <!-- IMPORTANT NOTE: Look for portions marked with the following:

         ***** FORECAST GROUP SPECIFIC ******

         Those sections must be modified if forecast groups
            are changed or added for generating
         lagged ensembles or executing MEFP. -->

    <!-- ============================================ -->
    <!--              Preprocess Data                 -->
    <!-- ============================================ -->
    <!-- Uncomment this activity if RFC forecast source is to be used for
         temperature forecasting.-->
    <!--<activity>
        <runIndependent>true</runIndependent>
        <workflowId>MEFP_Preprocess_RFC_Forecast</workflowId>
    </activity>-->
    <!-- <activity>
        <runIndependent>true</runIndependent>
        <workflowId>MEFP_Preprocess_GEFS_Forecast</workflowId>
    </activity>
    <activity>
        <runIndependent>true</runIndependent>
        <workflowId>MEFP_Preprocess_CFSv2_Forecast</workflowId>
    </activity> -->

    <!-- ============================================ -->
    <!--           Generate MEFP Forecast             -->
    <!-- ============================================ -->

    <!-- ****** FORECAST GROUP SPECIFIC ****** -->
    <!-- Execute the MEFP to generate FMAP and TFMN,
         TFMX forecasts -->
    <!-- Add one activity per group and data type. -->
    <activity>
        <runIndependent>true</runIndependent>
        <moduleInstanceId>FGroup_MEFP_FMAP_Forecast</moduleInstanceId>
    </activity>
    <activity>
        <runIndependent>true</runIndependent>
        <moduleInstanceId>FGroup_MEFP_TFMN_TFMX_Forecast</moduleInstanceId>
    </activity>

    <!-- Converts all TFMN/TFMX data to FMAT forecast data
    for all groups at once. -->
    <activity>
        <runIndependent>true</runIndependent>
        <moduleInstanceId>MEFP_FMAT_Forecast</moduleInstanceId>
        <ensemble>
            <ensembleId>MEFP</ensembleId>
            <runInLoop>true</runInLoop>
        </ensemble>
    </activity>

</workflow>
```

2.1.2 Set the MEFP_HINDCASTING Global Property

Action: Modify the global properties file for the *hindcasting SA*:

<*region_dir*>/sa_global.properties

Set the value of the property, MEFP_HINDCASTING, to be true:

```
MEFP_HINDCASTING=true
```

Description: The default delivered example module configuration files for the MEFPEnsembleGeneratorModelAdapter make use of the MEFP_HINDCASTING global property in order to define a run-file property with the name of hindcasting. If you configuration does not already make use of that run-file property, then the next step will including defining it appropriately.

2.1.3 Configure the MEFPEnsembleGeneratorModelAdapter for Hindcasting

Action: In the *hindcasting SA*, modify the module configuration files for any module that executes the MEFPEnsembleGeneratorModelAdapter and will be run as part of the HEFS hindcast.

For example, the following two such module configuration files are installed and modified as part of the default MEFP forecast component configuration (see *MEFP Configuration Guide: Forecast Components*, Sections 2.2.5 and 2.2.6):

<*configuration_dir*>/ModuleConfigFiles/hefs/FGroup/…
FGroup_MEFP_FMAP_Forecast.xml
FGroup_MEFP_TFMN_TFMX_Forecast.xml

The changes to make are as follows:

- **Ensure that the hindcasting run-file property is set to "$MEFP_HINDCASTING$"**
  For example:

  ```
  <string key="hindcasting" value="$MEFP_HINDCASTING$"/>
  ```

- **Define the forecast sources to use and set the number of forecast days appropriately**
  For example:

  ```
  <int key="rfcNumberOfForecastDays" value="0"/>
  <int key="gefsNumberOfForecastDays" value="15"/>
  <int key="cfsv2NumberOfForecastDays" value="0"/>
  <int key="climatologyNumberOfForecastDays" value="0"/>
  ```

Description: The hindcasting run-file property indicates to MEFP whether it should run in hindcast mode (true) or operational mode (false or undefined). If true (as set through the MEFP_HINDCASTING global property in the previous step), then MEFP will load input forcing time series for the used forecast sources from the location's parameter file.

The *NumberOfForecastDays properties define if a forecast source is included (value greater than 0) and how many days of the forecast source is used.

See the Example below, which is the default delivered FGroup_MEFP_FMAP_Forecast.xml module configuration file. It is setup for hindcasting using the MEFP_HINDCASTING global property and so that the forecast sources GEFS, CFSv2, and resampled climatology are each included for 15 days, 270 days, and 330 days, respectively.

EXAMPLE: Delivered FGroup_MEFP_FMAP_Forecast.xml

```
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://fews.wldelft.nl/schemas/version1.0/generalAdapterRun.xsd">
    <general>
        <description>MEFP Ensemble Generator</description>
        <piVersion>1.8</piVersion>
        <rootDir>%TEMP_DIR%</rootDir>
        <workDir>%ROOT_DIR%/work</workDir>
        <exportDir>%ROOT_DIR%/input</exportDir>
        <exportDataSetDir>%ROOT_DIR%</exportDataSetDir>
        <exportIdMap>IdExportMEFPMAP</exportIdMap>
        <importDir>%ROOT_DIR%/output</importDir>
        <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
        <dumpDir>%ROOT_DIR%</dumpDir>
        <diagnosticFile>%ROOT_DIR%/output/diag.xml</diagnosticFile>
    </general>
    <activities>

        <startUpActivities>
        </startUpActivities>

        <exportActivities>
            <exportTimeSeriesActivity>
                <exportFile>inputs.xml</exportFile>
                <timeSeriesSets>

                    <!-- RFC QPF FMAP (future MAP, in this case) time series;
                         uncomment if used. -->
<!--                    <timeSeriesSet>
                        <moduleInstanceId>FMAP_PreProcessing_QPF</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>MAPX</parameterId>
                        <locationSetId>Catchments_HEFS_FGroup</locationSetId>
                        <timeSeriesType>simulated forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="6"/>
                        <relativeViewPeriod unit="day" start="0" startOverrulable="true" end="5"
endOverrulable="false"/>
                        <readWriteMode>read only</readWriteMode>
                    </timeSeriesSet>
-->
                    <!-- GEFS FMAP Ensemble Mean -->
                    <timeSeriesSet>
                        <moduleInstanceId>MEFP_GEFS_Interpolate_Location_FMAP</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>FMAP</parameterId>
                        <qualifierId>GEFS</qualifierId>
                        <locationSetId>Catchments_HEFS_FGroup</locationSetId>
                        <timeSeriesType>external forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="6"/>
                        <relativeViewPeriod unit="day" start="0" startOverrulable="true" end="16"
endOverrulable="false"/>
                        <readWriteMode>read only</readWriteMode>
                    </timeSeriesSet>

                    <!-- CFSv2 entire lagged ensemble -->
                    <timeSeriesSet>

<moduleInstanceId>FGroup_MEFP_CFSv2_FMAP_LaggedEnsemble</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>FMAP</parameterId>
                        <qualifierId>CFSv2</qualifierId>
                        <locationSetId>Catchments_HEFS_FGroup</locationSetId>
                        <timeSeriesType>external forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="6"/>
                        <relativeViewPeriod unit="day" start="0" startOverrulable="true"
end="270" endOverrulable="false"/>
                        <readWriteMode>read only</readWriteMode>
                        <ensembleId>CFSv2</ensembleId>
                    </timeSeriesSet>

                </timeSeriesSets>
            </exportTimeSeriesActivity>

            <exportRunFileActivity>
                <exportFile>%ROOT_DIR%/run_info.xml</exportFile>
                <properties>
                    <int key="printDebugInfo" value="0"/>
                    <string key="hindcasting" value="$MEFP_HINDCASTING$"/>
                    <string key="parameterDir" value="$MEFP_ROOT_DIR$/mefpParameters"/>
                    <int key="rfcNumberOfForecastDays" value="0"/>
                    <int key="gefsNumberOfForecastDays" value="15"/>
                    <int key="cfsv2NumberOfForecastDays" value="270"/>
                    <int key="climatologyNumberOfForecastDays" value="330"/>
                    <string key="rfcUseEPT" value="true"/>
                    <string key="useResampledClimatology" value="true"/>
                    <int key="initialEnsembleYear" value="1961"/>
                    <int key="lastEnsembleYear" value="1997"/>
                    <string key="eptUseStratifiedSampling" value="true"/>
                </properties>
            </exportRunFileActivity>
        </exportActivities>

        <executeActivities>
            <executeActivity>
                <command>

<className>ohd.hseb.hefs.mefp.adapter.MEFPEnsembleGeneratorModelAdapter</className>
                    <binDir>$OHDBINDIR$</binDir>
                </command>
                <arguments>
                    <argument>%ROOT_DIR%/run_info.xml</argument>
                </arguments>
                <timeOut>300000</timeOut>
            </executeActivity>
        </executeActivities>

        <importActivities>
            <importTimeSeriesActivity>
                <importFile>outputs.xml</importFile>
                <timeSeriesSets>
                    <timeSeriesSet>
                        <moduleInstanceId>FGroup_MEFP_FMAP_Forecast</moduleInstanceId>
                        <valueType>scalar</valueType>
                        <parameterId>FMAP</parameterId>
                        <locationSetId>Catchments_HEFS_FGroup</locationSetId>
                        <timeSeriesType>external forecasting</timeSeriesType>
                        <timeStep unit="hour" multiplier="6"/>
                        <readWriteMode>add originals</readWriteMode>
                        <ensembleId>MEFP</ensembleId>
                    </timeSeriesSet>
                </timeSeriesSets>
            </importTimeSeriesActivity>
        </importActivities>

    </activities>
</generalAdapterRun>
```

2.2 *Configuration File Changes for Exporting Time Series (Required)*

Changes to make to export time series for verification by the EVS are as follows:

1. Configure export modules.
2. Add export module descriptors to ModuleInstanceDescriptors.xml.
3. Configure export workflow to execute export modules.
4. Add export workflow descriptor to WorkflowDescriptors.xml.
5. Create/Modify the hindcasting workflow.
6. Modify the hindcasting workflow descriptor.

Each step is described in the following sections.

Throughout this section, examples are provided using two operational basins (segments) at MARFC: WALN6DEL and CNNN6DEL. Both are in the UpperDelaware forecast group. WALN6DEL is a headwater basin and CNNN6DEL is its downstream basin.

2.2.1 Configure Export Modules

Action: For each location/basin/gauge for which you want to verify results, define an export module. The file must be placed in a subdirectory of <*configuration_dir*>/ModuleConfigFiles; which subdirectory is up to the user. For example, the following may be defined for WALN6DEL:

<*configuration_dir*>/ModuleConfigFiles/hefs/UpperDelaware/WALN6DEL_HEFS_Hindcast_Export.xml

The module configuration file should be for a generalAdapterRun and include an exportTimeSeriesActivity, but with all other activity XML elements left empty or not included. In the exportTimeSeriesActivity, add a timeSeriesSet XML element for each time series that is to be included in verification using EVS. For example, the following hindcasts may be verified:

- MEFP generated ensemble of precipitation
- MEFP generate ensemble of temperature
- Streamflow ensemble generated by executing the hydrologic model using MEFP generated forcings as input
- EnsPost prost-processed streamflow ensemble

The MEFP forcing exported should reflect whatever is used as input to the hydrologic models. Thus, if a merge transformation (e.g., MergeMAP*, or MergeMAT* modules) is used to combine the MEFP output with other forcing sources prior to running the hydrologic models, then the output from that merge transformation should be exported.

Description: Through the use of a generalAdapterRun, ensembles generated at various steps in the ensemble generation process may be exported. For use in EVS, it is recommended that:

One export file is created per *hindcast T0*, location (locationId), and data type (parameterId).

FEWS templates can be used to define <u>one</u> export module configuration file <u>per data type</u> that is executed <u>many times</u>, once for each location, to generate files that satisfy the requirement above. It does this in a workflow configuration file through the use of the properties XML element within an activity XML element. See Section 4.11.

The **TimeSeriesExporterModelAdapter**, delivered with HEFS and documented in the *MEFP User's Manual* (Section 4.2), allows for <u>one</u> module executed <u>only one time</u> to generate files for the current T0 and many locations and data types that satisfy the requirement above. It is a general adapter implementation and, therefore, has memory limitations related to the number of time series that can be processed through a single module execution. See the *MEFP Plugin Framework Configuration Guide* (Section 3.3.3) for an example of its use.

The examples below are of export module configuration files for the basin WALN6DEL. Examples of the following exports are provided:

- <T0>_WALN6DEL_FMAP.fi – Contains the MEFP generated ensemble of precipitation
- <T0>_WALN6DEL_FMAT.fi – Contains the MEFP generated ensemble of temperature
- <T0>_WALN6DEL_RAW_SQIN.fi – Contains the hydrologic model generate ensemble of streamflows created by using MEFP generated forcings as input
- <T0>_WALN6DEL_POSTPROCESSED_SQIN.fi – Contains the EnsPost post-processed streamflow ensemble.

In each example below, the export file name includes the *hindcast T0* as the first component of its file name, followed by the basin locationId and data type. The files generates are .fi/.bin format (i.e., the file name includes .fi as its extension and the exportBinFile XML element is set to true) so as to minimize file size.

EXAMPLE: WALN6DEL export module for MEFP generated FMAP ensemble

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>WALN6DEL HEFS FMAP Hindcast Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/hefs</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
            <exportFile>%TIME0%_WALN6DEL_FMAP.fi</exportFile>
            <exportBinFile>true</exportBinFile>
            <timeSeriesSets>
               <!-- MEFP Precipitation Export -->

               <timeSeriesSet>
<moduleInstanceId>UpperDelaware_MEFP_FMAP_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>FMAP</parameterId>
                  <locationId>WALN6DEL</locationId>
                  <timeSeriesType>external forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0" startOverrulable="false" end="7920"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start="1961" end="1997"/>
               </timeSeriesSet>
            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>
      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

EXAMPLE: WALN6DEL export module for MEFP generated FMAT ensemble

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>WALN6DEL HEFS FMAT Hindcast Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/hefs</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
            <exportFile>%TIME0%_WALN6DEL_FMAT.fi</exportFile>
            <exportBinFile>true</exportBinFile>
            <timeSeriesSets>
               <!-- MEFP Temperature Export -->

               <timeSeriesSet>
                  <moduleInstanceId>MEFP_FMAT_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>FMAT</parameterId>
                  <locationId>WALN6DEL</locationId>
                  <timeSeriesType>external forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0" startOverrulable="false" end="7920"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start="1961" end="1997"/>
               </timeSeriesSet>
            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

EXAMPLE: WALN6DEL export module for hydrologic model generated SQIN ensemble

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>WALN6DEL HEFS RAW SQIN Hindcast Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/hefs</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
            <exportFile>%TIME0%_WALN6DEL_RAW_SQIN.fi</exportFile>
            <exportBinFile>true</exportBinFile>
            <timeSeriesSets>

               <!-- Hydrologic Model Streamflow Export -->

               <timeSeriesSet>
<moduleInstanceId>ADDSUB_WALN6DEL_ADD.BFQ_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>SQIN</parameterId>
                  <locationId>WALN6TOT</locationId>
                  <timeSeriesType>simulated forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0" startOverrulable="false" end="7920"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start="1961" end="1997"/>
               </timeSeriesSet>
            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

EXAMPLE: WALN6DEL export module for EnsPost post-processed SQIN ensemble

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>WALN6DEL HEFS EnsPost SQIN Hindcast Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/hefs</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
            <exportFile>%TIME0%_WALN6DEL_POSTPROCESSED_SQIN.fi</exportFile>
            <exportBinFile>true</exportBinFile>
            <timeSeriesSets>

               <!-- EnsPost Export -->

               <timeSeriesSet>
                  <moduleInstanceId>WALN6DEL_ENSPOST_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>SQIN</parameterId>
                  <locationId>WALN6DEL</locationId>
                  <timeSeriesType>simulated forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="0" startOverrulable="false" end="7920"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>HEFSENSPOST</ensembleId>
                  <ensembleMemberIndexRange start="1961" end="1997"/>
               </timeSeriesSet>
            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

2.2.2 Add Export Module Descriptors to ModuleInstanceDescriptors.xml

Action: Add a module instance descriptor for each module created in the preceding step to the file:

<*configuration_dir*>/RegionConfigFiles/ModuleInstanceDescriptors.xml

Description: The modules must be added to the descriptors so that they can be executed within a workflow.

The example below defines module instance descriptors for the four modules used as an example in the previous step.

EXAMPLE: ModuleInstanceDescriptors.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<moduleInstanceDescriptors xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews
http://chps1/schemas/moduleInstanceDescriptors.xsd" version="1.0">

...

    <moduleInstanceDescriptor id="WALN6DEL_HEFS_Hindcast_FMAP_Export">
              <moduleId>GeneralAdapter</moduleId>
    </moduleInstanceDescriptor>
    <moduleInstanceDescriptor id="WALN6DEL_HEFS_Hindcast_FMAT_Export">
              <moduleId>GeneralAdapter</moduleId>
    </moduleInstanceDescriptor>
    <moduleInstanceDescriptor id="WALN6DEL_HEFS_Hindcast_RAW_SQIN_Export">
              <moduleId>GeneralAdapter</moduleId>
    </moduleInstanceDescriptor>
    <moduleInstanceDescriptor id="WALN6DEL_HEFS_Hindcast_EnsPost_SQIN_Export">
              <moduleId>GeneralAdapter</moduleId>
    </moduleInstanceDescriptor>

...

</moduleInstanceDescriptors>
```

2.2.3 Configure Export Workflow to Execute Export Modules

Action: Configure an appropriate workflow to execute the export modules created in the preceding step. For example, the following module may be defined for MARFC:

<*configuration_dir*>/WorkflowFiles/hefs/HEFS_Hindcast_Export.xml

Description: The workflow should execute the modules as necessary in order to export the time series for verification.

If the **FEWS template** approach is used, each activity must include a properties XML element, and there will be one activity defined per location and data type. See Section 4.11.

If the **TimeSeriesExporterModelAdapter** approach is used, then only a single activity must be defined to execute the one export module. That module then must include all time series to export for all locations and data types in the exportTimeSeriesActivity. If memory limitations cause problems, multiple calls can be made to the adapter as needed, each defined as separate modules.

In the example below, the four modules defined in the preceding step are executed.

EXAMPLE: WALN6DEL Export Workflow

```
<?xml version="1.0" encoding="UTF-8"?>
<workflow xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/workflow.xsd"
version="1.1">
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>WALN6DEL_HEFS_Hindcast_FMAP_Export</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>WALN6DEL_HEFS_Hindcast_FMAT_Export</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>WALN6DEL_HEFS_Hindcast_RAW_SQIN_Export</moduleInstanceId>
  </activity>
  <activity>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>WALN6DEL_HEFS_Hindcast_EnsPost_SQIN_Export</moduleInstanceId>
  </activity>

</workflow>
```

2.2.4 Add Export Workflow Descriptor to WorkflowDescriptors.xml

Action: Add a workflow descriptor for the workflow created in the preceding step to the file:

<*configuration_dir*>/RegionConfigFiles/WorkflowDescriptors.xml

Description: The workflow must be added to the descriptors so that it can be executed via other workflows.

The example below defines workflow descriptor for the workflow with id "HEFS_Hindcast_Export" created in the previous step.

EXAMPLE: ModuleInstanceDescriptors.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<workflowDescriptors xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/workflowDescriptors.xsd"
version="1.0">

...

    <workflowDescriptor id="HEFS_Hindcast_Export" name="HEFS Hindcast Export" forecast="false"
        visible="false" allowApprove="false" autoApprove="false">
        <description>HEFS Hindcast Export</description>
    </workflowDescriptor>

...

</workflowDescriptors>
```

2.2.5 Create/Modify the Hindcasting Workflow

Action: Identify the HEFS workflow that will be executed in order to generate the desired hindcasts. The workflow must include the following actions:

- Execute the MEFP for the desired locations to generate precipitation and temperature forcing ensembles.
- Execute the operational streamflow pre-processing component. That component, typically include merge transformations, combines the MEFP output with other sources of forcings in before executing the hydrologic models.
- Execute the hydrologic models to generate ensemble streamflow time series.
- Execute EnsPost to post-process the ensemble streamflow time series.

You may choose to create a new workflow for two reasons: no workflow exists that performs all actions above, or because the next step involves modifying the workflow and you do not want to modify an existing workflow. Whatever the reason, it is recommended you use an operational workflow as a starting point and modify it as needed to perform all actions above. Be sure to name the workflow appropriately and add an appropriate descriptor to the WorkflowDescriptors.xml configuration file.

Action: Add an activity XML element to execute the export workflow for which a descriptor was added in the preceding step.

You may choose to create a new workflow instead of modifying an existing workflow. To do so, copy the configuration file for the workflow identified above to give it a new (appropriate) name and add an appropriate descriptor to the WorkflowDescriptors.xml configuration file.

Action: Ensure that each activity in the workflow has its runIndependent XML element set to “false” so that the workflow will stop if an error occurs in the MEFP. This will prevent, for example, old MEFP output from being used as input to the hydrologic model and from being exported with the current *hindcast T0* associated with it.

Description: The export workflow should be executed after the HEFS components that generate its time series are executed. The example below shows a newly constructed workflow for hindcast generation using the workflowId “HEFS_Hindcast”.

The example below is for the UpperDelaware forecast group in MARFC, for which the pre-processing component is executed as part of the UpperDelaware_HEFS_Forecast workflow. The hindcasting workflow then includes the MEFP_Forecast workflow executed before it and the export workflow executed after.

EXAMPLE: HEFS_Hindcast.xml

```xml
<workflow xmlns="http://www.wldelft.nl/fews" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/workflow.xsd" version="1.1">

<!-- Already configured, Preprocessing is turned off -->
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>MEFP_Forecast</workflowId>
  </activity>

<!-- Already configured, and includes HEFS preprocessing -->
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>UpperDelaware_HEFS_Forecast</workflowId>
  </activity>

<!-- EnsPost Forecast is done in UpperDelaware_HEFS_Forecast -->

<!-- New for Hindcasting -->
  <activity>
    <runIndependent>false</runIndependent>
    <workflowId>HEFS_Hindcast_Export</workflowId>
  </activity>
</workflow>
```

2.2.6 Modify the Hindcasting Workflow Descriptor

Action: Modify the workflow descriptor for the *hindcasting workflow* (either an existing one or the one just created in the previous step).

If an existing workflow is modified, move its descriptor to the top of the file.

The workflow descriptor should be as follows (the id attributes match the workflowId used in the example of the previous section; modify it and the name as needed):

```
<workflowDescriptor id="HEFS_Hindcast" name="HEFS Hindcast" forecast="true" visible="true"
allowApprove="true" autoApprove="true">
  <description>HEFS Hindcast</description>
    <cardinalTimeStep id="12Z"/>
     <stateSelection>
            <warmState>
              <stateSearchPeriod start="-6" end="0" unit="day"/>
            </warmState>
     </stateSelection>
     <runExpiryTime unit="hour" multiplier="1"/>
</workflowDescriptor>
```

When copying or editing an operational workflow, set the end day of the warmState stateSearchPeriod to 0 (see green highlighted line above). The workflow will then use the most recent warm state.

It is also important to consider how long the hindcasts will remain in the localDataStore, which is dictated by the runExpiryTime XML element (see yellow highlighted line above). If the runExpiryTime is one hour as shown above, and the *hindcasting SA* is started with the rolling barrel on (fews_ohdPlugins.sh), then it will delete the hindcasts produced by HEFS_Hindcast when they are one hour old.

Description: The *hindcasting workflow* descriptor is setup to generate hindcasts at T0s of 12Z using only the appropriate states for each *hindcast T0*.

On the runExpiryTime:
If you run the hindcasting SA with rolling barrel on, then the warm states generated as part of the hindcasting process (see Section 3.2.2) may be deleted before the hindcasts are created. To avoid this, do one of the following:

- Run CHPS with the rolling barrel off (use the start script fews_ohdPlugins.sh.rboff).
- Recreate your warm states just before generating hindcasts.
- Equivalently, copy in a localDataStore with the warm states already prepared.
- Define an arbitrarily long runExpiryTime to your hindcast workflow:

```
<runExpiryTime unit="year" multiplier="1"/>
```

Also note that other workflows/modules called by your hindcast workflow may have their own expiry times set. This may cause problems if a module forecast with a runExpiryTime of one hour is deleted in the middle of a hindcast. All your workflows/modules should have an expiry time of at least as long as you expect your hindcast to run.

2.3 *Configuration File Changes for Other Modules As Needed*

Other configuration file changes that may need to be made are as follows:

1. Modify merge transformation for precipitation time series.
2. Add sample transformation for MAPE data type.
3. Modify the source of observations for the HEFSEnsPostModelAdapter.

Each step is described in the following sections.

Throughout this section, examples are provided using two operational basins (segments) at MARFC: WALN6DEL and CNNN6DEL. Both are in the “UpperDelaware” forecast group. WALN6DEL is a headwater basin and CNNN6DEL is its downstream basin.

2.3.1 Handling Merged Precipitation Time Series (Optional)

If the hydrologic models, SNOW17 in particular, use multiple sources of precipitation, such as MAP and MAPX, then perform this step.

Action: Modify any merge transformation, either used in updating states or operational ensemble generation, that prepares precipitation input for the hydrologic models so that any type of data included in the merge that is known to be missing is removed (or commented out) from that merge.

Description: If missing data is included in the merge transformation, FEWS will substitute zeros for those missing values. Thus, if MAP is missing and is merged before MAPX, which is non-missing, then resulting time series will always be zero because FEWS will replace the mssing MAP values with zero and then not use the MAPX time series. This problem can be avoided by removing or commenting out components from the merge that are known to be missing.

EXAMPLE: Merge transformation associated with an “UpdateStates” Workflow

Suppose the merge transformation used the following merge XML element:

```
<merge>
  <simple>
    <inputVariable>
      <variableId>LL_MAP_6</variableId>
    </inputVariable>
    <inputVariable>
      <variableId>LL_MAPX_6</variableId>
    </inputVariable>
    <moduleParameterFile>MERGETS_LL_MAP_UpdateStates</moduleParameterFile>
    <outputVariable>
       <variableId>LL_MAPX_6</variableId>
    </outputVariable>
  </simple>
</merge>
```

Then, if the timeSeriesSet defined in the module configuration file for variable LL_MAP_6 is known to be missing when generating a hindcast, then comment it out so that it is no longer included:

```
<merge>
  <simple>
    <!-- <inputVariable>
      <variableId>LL_MAP_6</variableId>
    </inputVariable> -->
    <inputVariable>
```

```
      <variableId>LL_MAPX_6</variableId>
    </inputVariable>
    <moduleParameterFile>MERGETS_LL_MAP_UpdateStates</moduleParameterFile>
    <outputVariable>
      <variableId>LL_MAPX_6</variableId>
    </outputVariable>
  </simple>
</merge>
```

2.3.2 Handling MAPE Time Series (Optional)

If as part of your standard (climatology) ESP forecast, you sample MAPE time series to be used as input to a hydrologic model, typically SAC-SMA, perform this step so that the same sampling is done for an HEFS ensemble.

Action: To the HEFS *hindcasting workflow*, add the necessary modules so that the MAPE is sampled, as it is for your standard ESP forecast, but assigned an ensembleId of “MEFP”.

Description: If MAPE is sampled for generating an ESP forecast, then any models that use that data will not work when executed using MEFP generated forcings because those use a different ensembleId (“MEFP”). Thus, new modules must be defined so that an MAPE ensemble is sampled with an ensembleId of “MEFP”.

EXAMPLE: Merge transformation associated with an “UpdateStates” Workflow

For MARFC, MAPE time series are sampled when generating an ESP forecast. To add the same sampling for use in generating HEFS hindcasts, the following was done.

The appropriate SampleESP* module was added as an activity to the pre-processing workflow executed in the *hindcasting workflow*. In this case, the workflow had the name

<*configuration_dir*>/WorkflowFiles/UpperDelaware/UpperDelaware_Preprocessing_Forecast.xml

The following activity XML elements were added:

```
<activity>
        <runIndependent>false</runIndependent>
        <moduleInstanceId>SetTimes_ESP_Forecast</moduleInstanceId>
</activity>
<activity>
        <runIndependent>false</runIndependent>
        <moduleInstanceId>UpperDelaware_SampleESP_Forecast</moduleInstanceId>
</activity>
```

The *SampleESP* module was modified so that the MAPE_Historic variable used a readWriteMode of “read complete forecast” and the output variable was modified to use an ensembleId of “MEFP”. The MAPE_Historic variable was set to use a readWriteMode of “read complete forecast” in order to ensure the entire available MAPE time series is read each time for sampling. For example, the following variable:

```
<variable>
    <variableId>MAPE_Historic</variableId>
    <timeSeriesSet>
            <moduleInstanceId>ImportSHEF</moduleInstanceId>
```

```
                <valueType>scalar</valueType>
                <parameterId>MAPE</parameterId>
                <locationSetId>Meteo_UpperDelaware</locationSetId>
                <timeSeriesType>external historical</timeSeriesType>
                <timeStep id="12Z"/>
                <relativeViewPeriod unit="day" start="-36500" end="0"/>
                <readWriteMode>read only</readWriteMode>
        </timeSeriesSet>
    </variable>
```

was modified to (note the shaded line):

```
    <variable>
        <variableId>MAPE_Historic</variableId>
        <timeSeriesSet>
                <moduleInstanceId>ImportSHEF</moduleInstanceId>
                <valueType>scalar</valueType>
                <parameterId>MAPE</parameterId>
                <locationSetId>Meteo_UpperDelaware</locationSetId>
                <timeSeriesType>external historical</timeSeriesType>
                <timeStep id="12Z"/>
                <relativeViewPeriod unit="day" start="-36500" end="0"/>
                <readWriteMode>read complete forecast</readWriteMode>
        </timeSeriesSet>
    </variable>
```

The output ensembleId was set to “MEFP” so that the hydrologic models executed within the HEFS *hindcasting workflow* would be provided the resulting sampled MAPE time series. That workflow is designed using the ensembleId is “MEFP” for all forcings (MAP, MAT, MAPE) and, therefore, the streamflow ensemble before EnsPost is applied. For example, the ensembleId XML element was modified as follows:

```
        <ensembleId>ESP</ensembleId>
```

was changed to:

```
        <ensembleId>MEFP</ensembleId>
```

2.3.3 Modify Observed Time Series in HEFSEnsPostModelAdapter (Optional)

Perform this step if either the ERD or ERS error model is applied in the execution of the HEFS ensemble post-processor (EnsPost). The model used is defined by the errorModel *run file property specified in the* exportRunFileActivity.

Action: Identify the timeSeriesSet XML element configured in the exportTimeSeriesActivity that defines the observed time series to be used by the HEFS EnsPost in order to apply an error model. Modify it to reference historical observations instead.

Description: When executed operationally, the observed streamflow is acquired from recent observations (relative to the current time), often contained within a QINE time series that is defined within the timeSeriesSet XML element. When hindcasting for a past date that may be years before the current time, those observations may not be available.

Thus, when hindcasting, the observed data used by EnsPost must be extracted from historical observations that have been imported into the CHPS localDataStore prior to executing EnsPost as part of hindcasting. Typically, the data is 24-hour QME data imported from datacard files, but it can also be 1-hour/6-hour QIN or QME and can be imported from other sources.

If no import is defined for historical observed streamflow, then make whatever changes are necessary to CHPS in order to import the historical observed streamflow and refer to the new import in the HEFSEnsPostModelAdapter configuration.

EXAMPLE: EnsPost Observed Time Series

The following example is based on an NCRFC configuration for the segment AEMI4. Suppose the exportTimeSeriesActivity is defined as follows:

```
<exportTimeSeriesActivity>
   <exportFile>inputs.xml</exportFile>
   <timeSeriesSets>
      <timeSeriesSet>
         <moduleInstanceId>ADJUSTQ_AMEI4_AMEI4_Forecast</moduleInstanceId>
         <valueType>scalar</valueType>
         <parameterId>QINE</parameterId>
         <locationId>AMEI4</locationId>
         <timeSeriesType>simulated forecasting</timeSeriesType>
         <timeStep unit="hour" multiplier="6"/>
         <relativeViewPeriod unit="hour" start="-120" startOverrulable="true" end="0"/>
         <readWriteMode>read only</readWriteMode>
      </timeSeriesSet>
      <timeSeriesSet>
         <moduleInstanceId>UNITHG_AMEI4_AMEI4_Forecast</moduleInstanceId>
```

```
            <valueType>scalar</valueType>
            <parameterId>SQIN</parameterId>
            <locationId>AMEI4</locationId>
            <timeSeriesType>simulated forecasting</timeSeriesType>
            <timeStep unit="hour" multiplier="6" />
            <relativeViewPeriod unit="hour" start="0" end="120" endOverrulable="true"/>
            <readWriteMode>add originals</readWriteMode>
            <ensembleId>MEFP</ensembleId>
            <ensembleMemberIndexRange start="1985" end="2010"/>
        </timeSeriesSet>
    </timeSeriesSets>
</exportTimeSeriesActivity>
```

The observed time series is the first timeSeriesSet in the timeSeriesSets XML element defined for the working segment (highlighted in yellow). In the example, above, the time series is acquired from an ADJUST-Q operation, which prepends observations to the beginning of simulated streamflow time series resulting in a time series with parameterId QINE.

It is possible for EnsPost to be configured to execute for multiple segments in a single configuration. In that case, the first time series defined for any segment is assumed to be the observed time series.

For hindcast dates well before T0, the ADJUST-Q operation will not find any observations to prepend, so no data before T0 will be included in the QINE time series. Thus, the observed data must be extracted from historical observed streamflow. In example here, the following import datacard module yielding 24-hour QME was used in place of the operational observed time series:

```
<timeSeriesSet>
    <moduleInstanceId>ImportDataCard</moduleInstanceId>
    <valueType>scalar</valueType>
    <parameterId>QME</parameterId>
    <locationId>AMEI4</locationId>
    <timeSeriesType>external historical</timeSeriesType>
    <timeStep multiplier="24" timeZone="GMT-6" unit="hour" />
    <relativeViewPeriod unit="hour" start="-120" startOverrulable="true" end="0"/>
    <readWriteMode>read only</readWriteMode>
</timeSeriesSet>
```

3 Generating Hindcasts

With the configuration changes complete, generating hindcasts is a three-step process:

- Import historical data needed to generate warm states.
- Generate historical warm states for all desired *hindcast T0*s.
- Generate hindcasts using those warm states.

Each step is described below.

To begin, do the following:

Start the hindcasting SA using your preferred technique. For example to start it with rolling barrel turned off and assuming that the ohdPlugins directory is properly located next to the SA's <*region_dir*>, use the command,

```
./ohdPlugins/fews_ohdPlugins.sh.rboff ##rfc_sa &
```

3.1 *Hindcast Example Information*

All sections below will use the following example as an illustration of hindcasting:

RFC: MARFC
Forecast Group: Upper Delaware
Segments: WALN6DEL and CNNN6DEL

Hindcast Period:
  Start: Jan 1, 1979
  End: Jan 1, 1999
Spin-up (Warmup) Period:
  Length: 2-years
  Start: Jan 1, 1977
Days Between Hindcasts: One day (a hindcast generated each day at 12Z)
Hindcast Length: 330 days

Update States Workflow Name: UpperDelaware_UpdateStates
Hindcast Workflow Name: HEFS Forecast

3.2 *Hindcast Generation Steps (Required)*

The steps of a hindcast are described in the following subsections.

3.2.1 Import Historical Data

Action: Identify sources of historical observed MAP, MAT, and QME/QIN data. If MAPE time series must also be employed (see Section 2.3.2), then identify a source of historical MAPE data. The source, typically datacard files for MAP and MAT, must be in a format that CHPS is configured to import.

Historical data is required on a 12Z forecasting clock of {18Z,12Z,0Z,6Z}. For historical data recorded in a local time system other than CST (UTC-6), a time shift is needed to map the data to the closest 12Z forecasting clock hour-of-the-day. In practice, the closest time for all RFCs is UTC-6 or CST. If your RFC does not already use CST, then modify the CHPS import configuration file appropriately.

For example, for datacard data, modify the file:

<*configuration_dir*>/ModuleConfigFiles/import/ImportDataCard.xml

and change the XML element timeZoneOffset from

```
<timeZoneOffset>+00:00</timeZoneOffset>
```

to

```
<timeZoneOffset>-06:00</timeZoneOffset>
```

wherever you find it in the file.

Action: Place the source files into the appropriate import subdirectory for the *hindcasting SA*. For example, the import subdirectory typically used for datacard data is as follows:

<*configuration_dir*>/Import/cardfiles

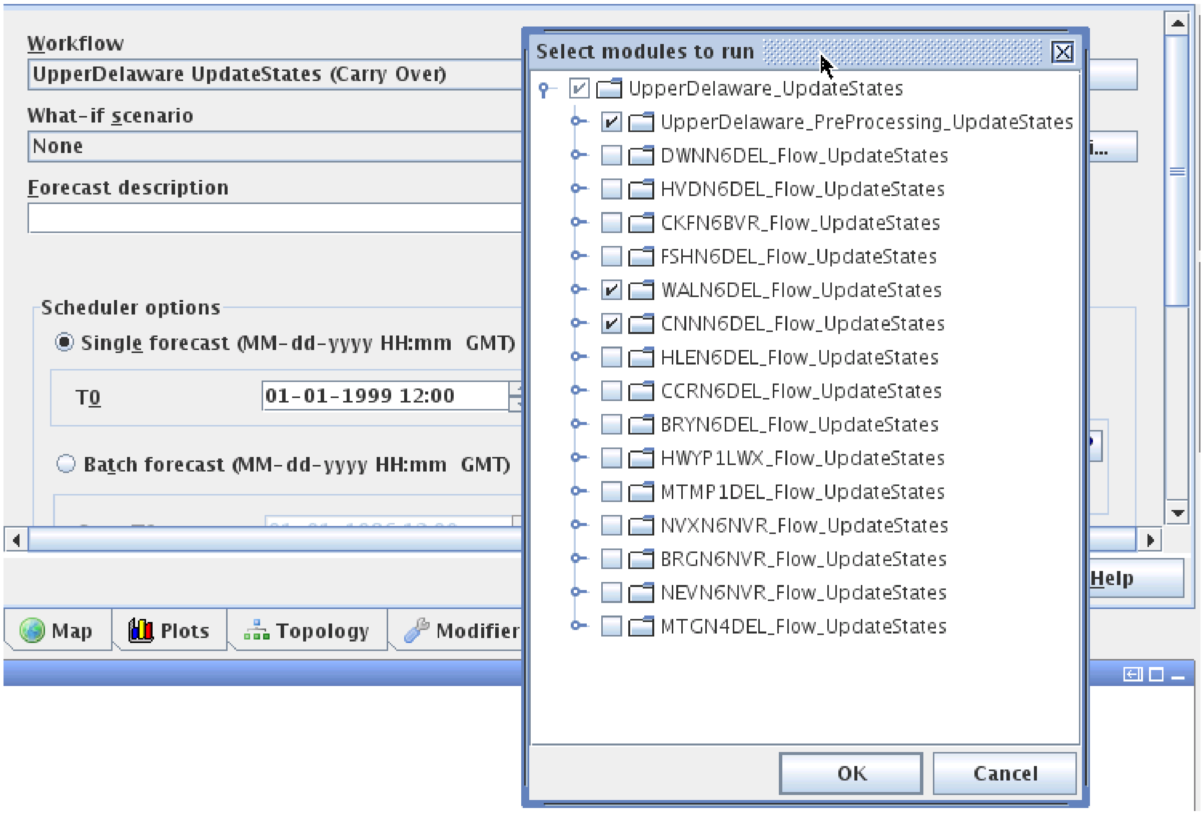
Action: Import the data by executing the appropriate workflow via the **Manual Forecast Dialog**. In most cases, that workflow is called “ImportScalars”.

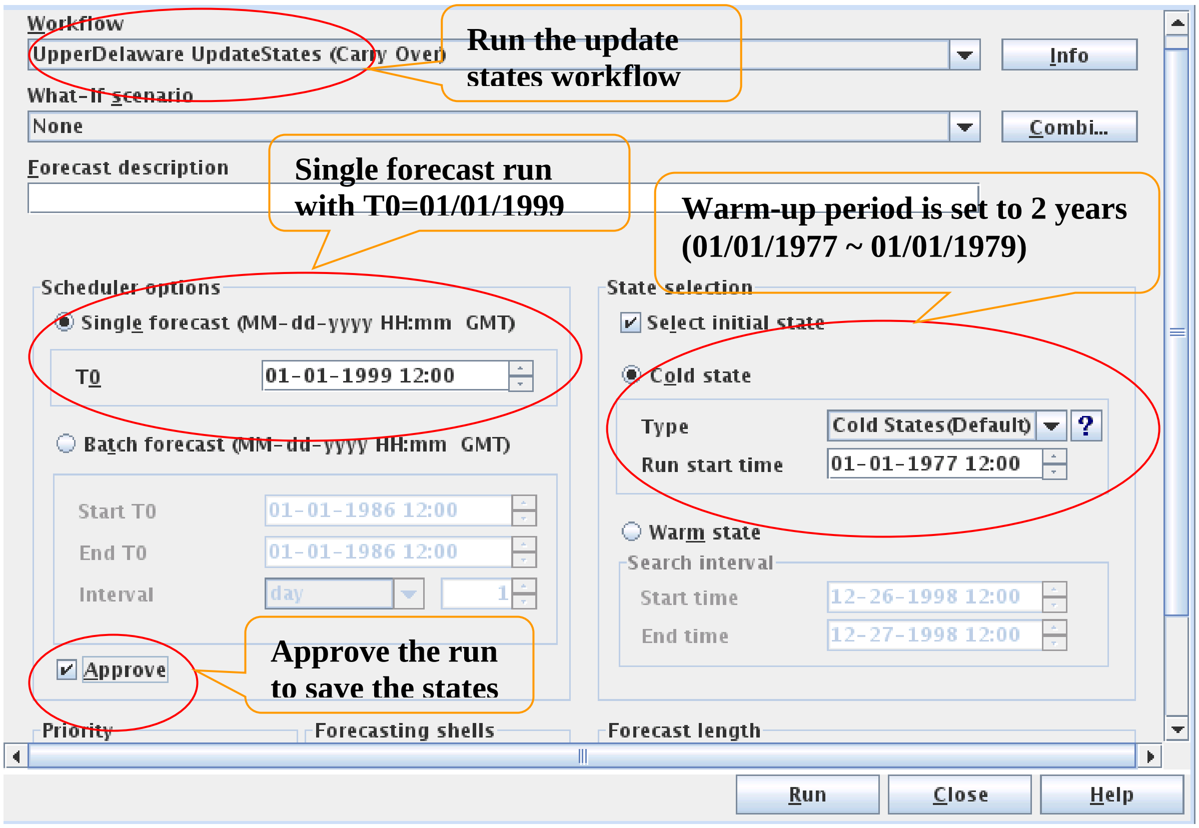
Description: Warm states must be generated for the hydrologic models in order generate HEFS hindcasts. To do so, *UpdateStates* workflow is run using historical observed forcing data as input. That data must be imported.

3.2.2 Generate Warm States

Action: Execute the appropriate *UpdateStates* workflow to generate historical warm states.

The example below follows the description provided in Section 3.1.

#	Action
1	Select ‘UpperDelaware UpdateStates’ in the **Manual Forecast Display**. Restrict the run to only include the modules necessary in order to generate the needed states. In this example, only the segmetns WALN6DEL and CNNN6DEL are needed:  Workflow: UpperDelaware UpdateStates (Carry Over) What-if scenario: None Forecast description: Scheduler options ● Single forecast (MM-dd-yyyy HH:mm GMT) T0 01-01-1999 12:00 ○ Batch forecast (MM-dd-yyyy HH:mm GMT) Select modules to run ☑ UpperDelaware_UpdateStates ☑ UpperDelaware_PreProcessing_UpdateStates □ DWNN6DEL_Flow_UpdateStates □ HVDN6DEL_Flow_UpdateStates □ CKFN6BVR_Flow_UpdateStates □ FSHN6DEL_Flow_UpdateStates ☑ WALN6DEL_Flow_UpdateStates ☑ CNNN6DEL_Flow_UpdateStates □ HLEN6DEL_Flow_UpdateStates □ CCRN6DEL_Flow_UpdateStates □ BRYN6DEL_Flow_UpdateStates □ HWYP1LWX_Flow_UpdateStates □ MTMP1DEL_Flow_UpdateStates □ NVXN6NVR_Flow_UpdateStates □ BRGN6NVR_Flow_UpdateStates □ NEVN6NVR_Flow_UpdateStates □ MTGN4DEL_Flow_UpdateStates OK Cancel Help Map Plots Topology Modifier

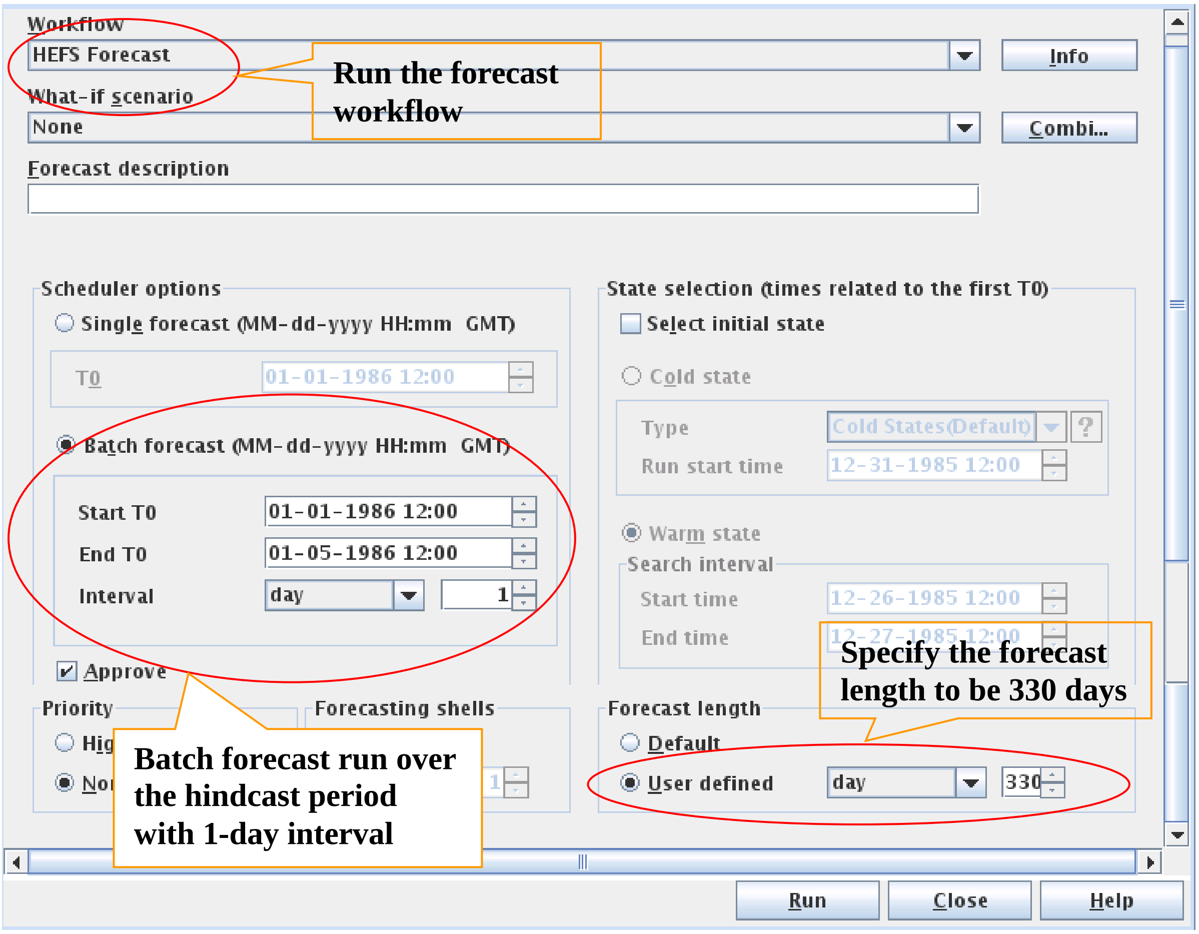
#	Action
2	Execute the workflow as shown below: - Set the **Schedule options** to be a "Single forecast" with a T0 of the last day of the hindcast period. - Set the **State selection** to be a "Cold state" with a "Run-start time" of the spin-up period start time (see Section 3.1) - Check the **Approve** checkbox.  Workflow: UpperDelaware UpdateStates (Carry Over) What-if scenario: None Forecast description: Scheduler options ● Single forecast (MM-dd-yyyy HH:mm GMT) T0 01-01-1999 12:00 ○ Batch forecast (MM-dd-yyyy HH:mm GMT) Start T0 01-01-1986 12:00 End T0 01-01-1986 12:00 Interval day 1 ☑ Approve State selection ☑ Select initial state ● Cold state Type Cold States(Default) Run start time 01-01-1977 12:00 ○ Warm state Search interval Start time 12-26-1998 12:00 End time 12-27-1998 12:00 Priority Forecasting shells Forecast length Run the update states workflow Single forecast run with T0=01/01/1999 Warm-up period is set to 2 years (01/01/1977 ~ 01/01/1979) Approve the run to save the states
3	Click the **Run** button.

Upon completion, the *hindcast SA* will have warm states stored in its localDataStore for all hindcast dates specified in Section 3.1.

3.2.3 Generate Hindcasts

Action: Execute the appropriate *UpdateStates* workflow to generate historical warm states.

The example below follows the description provided in Section 3.1.

#	Action
1	Execute the workflow as shown below: • Select the appropriate workflow: “HEFS Forecast”. • Set the **Schedule options** to be a “Batch forecast” with a Start T0, End T0, and Interval matching those described in Section 3.1. • Set the **Forecast length** to be “User defined” and 330 days.  Workflow: HEFS Forecast What-if scenario: None Forecast description: Scheduler options ○ Single forecast (MM-dd-yyyy HH:mm GMT) T0 01-01-1986 12:00 ● Batch forecast (MM-dd-yyyy HH:mm GMT) Start T0 01-01-1986 12:00 End T0 01-05-1986 12:00 Interval day 1 ☑ Approve Priority ○ Hig ● Nor Forecasting shells 1 State selection (times related to the first T0) □ Select initial state ○ Cold state Type Cold States(Default) Run start time 12-31-1985 12:00 ● Warm state Search interval Start time 12-26-1985 12:00 End time 12-27-1985 12:00 Forecast length ○ Default ● User defined day 330 Run the forecast workflow Batch forecast run over the hindcast period with 1-day interval Specify the forecast length to be 330 days
2	Click the **Run** button. While executing, after each T0, appropriate files will be exported as per the export modules defined added as part of the steps in Section 2.2. Additionally, the **CHPS Database Viewer** can be used to view the results as they are generated for each *hindcast T0*.

3.3 *Hindcast Date Restrictions*

Hindcasts can only generated for those dates for which (1) warm states can be created and (2) the used MEFP forecast sources have reforecasts or archived forecasts. The range of available dates for the default forecast sources are as follows:

Source	Used by	Years
RFC	MEFP	Varies by RFC (< 10 years, typically)
GEFS	MEFP	1985-2012 (last T0 is 7/30/2012 12Z)
CFSv2	MEFP	1982-2010 (T0 every 5 days)
MAT/MAP (historical observed temperature and precipitation data)	MEFP, EnsPost	varies by RFC
SQIN (historical simulated streamflow)	EnsPost	varies by RFC
QME/QIN/QINE (observed streamflow)	EnsPost	varies by RFC

3.3.1 Incorporating CFSv2 – The Leap Year Problem

As noted in the previous section, CFSv2 is available for T0s every 5 days. Specifically, for each year, the first reforecast is available on January 1 of that year and then every 5 days thereafter, but **ignoring leap days**. FEWS, however, includes leap days in its batch scheduler. So, to get consistent results when the hindcasting period crosses leap days, a series of 5-day, leap-day avoiding hindcasts is recommended.

For example, suppose hindcasts are to be generated including CFSv2 from Jan 1, 1985, through Dec 31, 1998. In the **Manual Forecast Dialog**, the forecast interval must be set to 5 days and the forecast length set appropriately to whatever length is desired. To avoid the leap days, the hindcasts would need to be generated through four separate batch runs (i.e., enter the start/end date range below and press the **Run** button after each one)

Start T0 = 1985-01-01, End T0 =1988-02-28
Start T0 = 1988-03-02, End T0 =1992-02-28
Start T0 = 1992-03-02, End T0 =1996-02-28
Start T0 = 1996-03-02, End T0 =1998-12-31

The CHPS batch forecaster schedules and runs forecasts such that you do not need to wait for the previous batch to complete before entering the new batch. Thus, for the above example, each batch can be specified in the **Manual Forecast Dialog** and the **Run** button clicked immediately.

4 Tips

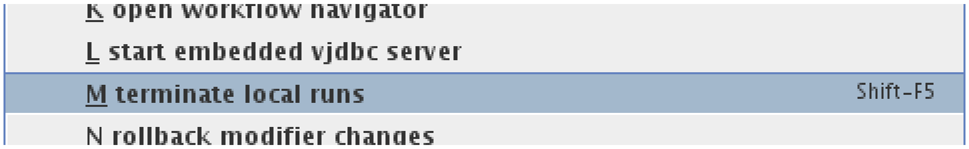
This section provides general tips for generating HEFS hindcasts.

4.1 *Keep the Hindcasting SA Configuration Current*

The *hindcasting SA* configuration should be updated with all operational configuration changes, especially changes to baseline CHPS.

4.2 *Terminating a Hindcast*

Terminating a hindcast can be accomplished using the standard CHPS tool to terminate all local runs. In the CHPS **Logs** panel, press F12 and select M, "terminate local runs":


K open workflow navigator
L start embedded vjdbc server
M terminate local runs
Shift-F5
N rollback modifier changes

4.3 *Turning off the Screensaver*

During hindcasts, the screen saver may disrupt the display and stop a hindcast run. To avoid this potential problem, disable the screen saver. Talk to the RFC IT person for how to do so.

4.4 *Using the check_exports.sh Script*

If the export files generated during hindcasting obey the naming conventions indicated by the examples provided in Section 2.2.1, then a check_exports.sh script can be used to identify *hindcast T0*s for which a hindcast failed to generate. It will also create a FEWS macro that allows for generating hindcasts only for those *hindcast T0*s and may be used to identify when old time series were incorrectly exported.

See FogBugz 1735 for more information:

http://schuylkill.nws.noaa.gov:7069/default.asp?1735.

4.5 *Checking that Time Series are Being Exported*

In addition to using the **Database Viewer** to verify that hindcasts are executing successfully (see Section 3.2.3), the export files can be examined.

The following presumes that you are exporting all files to be below a specific directory, referred to as the <*export_root_dir*>. Furthermore, all files are being exported satisfying the condition stated in Section 2.2.1: that they are *hindcast T0*, location, and parameter specific.

Do the following:

```
cd <export_root_dir>
ls –lrt
```

The time stamps of the files listed will provide an estimate of how long it takes to generate a hindcast for each *hindcast T0*.

If the data is exported to .xml files, as opposed to .fi or .fi/.bin files (as is done for the examples in this configuration guide), then grep and wc Linux commands can be used to count the number of missing values output to each file. Execute the following from the command line (copy-and-paste):

```
for i in *; do
echo
echo -n "$i not missing:"
grep event $i | grep -v 999 | wc -l
echo -n "$i     missing:"
grep event $i | grep    999 | wc -l
done
```

The resulting word-counts indicate the number of missing values found. Missing values may indicate an error in generating hindcasts.

- The example modules provided in Section 2.2.1 generate .fi/.bin files and cannot be used in conjunction with the for-command above.
- The for-command assumes that the missing value is set to "-999" for all exported files. See the missVal subelement definition in the general XML element within the examples of Section 2.2.1.

4.6 *Investigating Failures for Hindcast T0*

To investigate why a T0 failed, examine the resulting log.txt file created by FEWS:

<*region_dir*>/log.txt

When a CHPS batch forecast is run, the T0s are scheduled (dispatched), and then they are executed. If a hindcast is not generated for a specific *hindcast T0*, then it is either because it was not scheduled, or (more likely) it was scheduled and the run failed due to a data error.

Assume a hindcast failed to generate for the T0 1991-06-15.

1. Check If It Was Scheduled

Look in the log.txt file for a line that contains “1991-06-15” and text similar to the following:

```
INFO - ManualForecastDialog.dispatchTasks - ManualForecast.Started: Manual forecast
```

To do this, perform the following commands:

```
cd <region_dir>
grep -nr '06-15-1991' log.txt
```

Dispatching is done early in the batch process and the grep command should return quickly.

2. Check Whether a Scheduled Task Successfully Executed

First look for the T0 in log.txt within a message similar to the following:

```
TaskRun.Started: Starting Task HEFS_Hindcast with ID 2366 and T0 1991-06-15 12:00:00 and forecast
length of 15 days
```

Next, to decrease the number of lines that must be examined, extract the log.txt file lines after the “TaskRun.Started” text found above through the use of the –A option for grep, and write it to a second file. The following command will extract 10,000 lines of text and create the file 1991-06-15.txt containing those lines:

```
grep -A10000 'and T0 1991-06-15 12:00:00 and forecast length' log.txt > 1991-06-15.txt
vi 1991-06-15.txt
```

The number of lines being 10,000 was selected to ensure that <u>all</u> log.txt lines corresponding to the desired hindcast T0 are extracted. It may actually be bigger than necessary. Select a fewer number of lines if desired.

With all lines for the desired *hindcast T0* extracted, look in the 1991-06-15.txt file for “ERROR”. For example, this T0 hit a timeout of 5 minutes, possibly due to a high machine load:

```
[2014-01-27 01:33:14,728] ERROR - WorkflowPluginActivity$1.call -
ohd.hseb.ohdfewsadapter.OHDFewsAdapter failed with error Execution of Java class
ohd.hseb.ohdfewsadapter.OHDFewsAdapter terminated after time out of 300 seconds while running
ExecuteJavaActivity: /awips/chps/ohd/3.3.a/bin ohd.hseb.ohdfewsadapter.OHDFewsAdapter "arguments:
/tmp/FEWS_CNRFC_SNOW17_WBGC1H_WBGC1HUF_Forecast_MEFP_1977_9062953735510577241/r
un_info.fi " for ensemble MEFP_1977
nl.wldelft.fews.system.plugin.generaladapter.GeneralAdapterException: Execution of Java class
ohd.hseb.ohdfewsadapter.OHDFewsAdapter terminated after time out of 300 seconds version:
NWS2013.01, build: 43108 Nov7, type: stable, jre: 1.7.0_11, mx: 954m, db: Firebird, region: cnrfc_sa+
```

The reported error message should help in diagnosing the cause of the failure, as it would during any operational execution of HEFS.

4.7 *Speeding up a Hindcast Using Parallel Processors*

If you have more than one processor core available, the temperature potion of the MEFP run can be parallelized using the FEWS Multi-core option (FogBugz issue 1150). To count the number of available processors, execute the following:

```
cat /proc/cpuinfo
```

and look for lines similar to the following:

```
processor      : 3
vendor_id      : GenuineIntel
```

This machine has 4 processors (0 – 3). To utilize all 4 processors, do the following:

1. In the workflow that executes the MEFPEnsembleGeneratorModelAdapter to generate temperature ensembles (in the release, the module uses id MEFP_Forecast), modify the activity that executes the module which yields 6-hour FMAT data from the TFMN/TFMX time series output by the MEFPEnsembleGeneratorModelAdapter (in the release, the module uses id MEFP_FMAT_Forecast). For example:

   ```
   <activity>
      <runIndependent>true</runIndependent>
      <moduleInstanceId>MEFP_FMAT_Forecast</moduleInstanceId>
      <ensemble>
         <ensembleId>MEFP</ensembleId>
         <runInLoop>true</runInLoop>
      </ensemble>
   </activity>
   ```

   In the module whose activity was modified above, comment all lines specifying an ensembleId. For example:

   ```
   <!-- <ensembleId>MEFP</ensembleId> -->
   ```

2. **If you are running a single SA on your machine**, add the following line to the file <*region_dir*>/sa_global.properties:

   ```
   runInLoopParallelProcessorCount=100
   ```

   Even though this machine has only 4 processors, Deltares recommends setting it to 100 to use the max available.

   If you are running multiple hindcasting SAs on your machine, set the following in each of their sa_global.properties:

```
runInLoopParallelProcessorCount=2
```

This directs each SA to use no more than 2 processors, one for CHPS, and one for the system. The SAs will not compete for processors. It is also recommended that each SA have a separate temporary working directory and a separate piServicePortNumber. So, for example, the sa_global.properties file for SA-1 would use the following properties:

```
tempDir=/tmp/hindcast_1
piServicePortNumber=8101
runInLoopParallelProcessorCount=2
```

and the sa_global.properties file for SA 2 would add:

```
tempDir=/tmp/hindcast_2
piServicePortNumber=8102
runInLoopParallelProcessorCount=2
```

The directories corresponding to the property tempDir for each SA must be created. If the different SAs do not use different tempDir directories, the one SA may be forced to wait for another SA when it is writing to the tempDir directory, leading to timeouts which can cause missing T0s (see Section 4.4). For run speed, the temporary directories should be on a disk local to where the SA is running (usually under /tmp).

As a check, the following command can be executed to determine the number of processor being used on the system on which hindcasts are being generated:

```
ps -e -o psr= | sort | uniq | wc -l
```

It will return a number indicating the number of processors in use.

4.8 *Avoiding Running Out of Diskspace*

One file, log.txt (CHPS log), and 2 directories, the localDataStore (CHPS database) and the <*export_root_dir*> directory (where hindcast export files are written) will grow during your hindcast. For example, after 10 years of a daily hindcast that included 29 MEFP locations and 14 streamflow gauges, two of which included HEFS EnsPost, the following was observed:

- log.txt increased by 7 GB
- localDataStore increased by 38 GB
- <*export_root_dir*> increased by 71 GB

To monitor their sizes, use du command:

```
du -sh <region_dir>/log.txt
du -sh <region_dir>/localDataStore
du –sh <export_root_dir>
```

To reduce the size of log.txt, edit Log4jConfig.xml in your SA directory. The following is an example of the root section at the bottom of the file:

```
<root>
  <priority value="INFO"/>
  <appender-ref ref="defaultLogFile"/>
  <appender-ref ref="splashScreen"/>
  <appender-ref ref="explorerLogPanel"/>
  <appender-ref ref="explorerAcknowladgeIcon"/>
  <appender-ref ref="dataStoreLogEntriesTable"/>
</root>
```

The priority XML element can be modified as follows:

Priority value…	**Will write…**
ERROR	only ERROR messages
WARN	only WARN and ERROR messages
INFO (default)	INFO, WARN, and ERROR messages, including hydrologic model output

Without INFO-level messages (i.e., a priority of ERROR or WARN), it can be difficult to tell if a run for a *hindcast T0* has started or ended. You can use the top command to see if FEWS is running and ls –lrt to check the export file times, as per Section 4.4.

If you have additional disk space available, you can move the contents of your localDataStore and <*export_root_dir*> to the directory and link to it before running (log.txt cannot be moved). For example, for the localDataStore:

```
cd $region_dir
ln -s /big_disk/localDataStore localDataStore
```

However, if that extra disk space is not on the local disk relative to the machine executing the hindcasting SA, it may require extra time to generate the hindcasts due to I/O time writing the export files to a shared drive.

To save disk space, it is recommend that files be exported to <*export_root_dir*> in the .fi/.bin file format.

4.9 *Increasing Java Memory Allocation*

The maximum size of the Java memory allocation pool (-Xmx) is set in the script used to start CHPS. For example, ./ohdPlugins/fews_ohdPlugins.sh.rboff may use the settings:

```
... -XX:MaxPermSize=128m -Xmx1024M $JVM_OPTS ...
```

The maximum size of the memory pool can be increased by modifying the –Xmx flag; for example:

```
... -XX:MaxPermSize=128m -Xmx2048M $JVM_OPTS ...
```

The -Xmx value should be a multiple of 1024.

The Java virtual machine (JVM) can be executed as a server instead of a client (the default). To do so, add -server:

```
... -XX:MaxPermSize=128m -Xmx2048M -server $JVM_OPTS ...
```

The *hindcasting SA* can startup faster if the minimum size of the Java memory allocation pool (_Xms) is set to match the -Xmx value. For example:

```
... -XX:MaxPermSize=128m –Xms2048M -Xmx2048M -server $JVM_OPTS ...
```

The *hindcasting SA* will start with 2 GB of memory already allocated.

To check the settings were applied successful, execute the following:

```
ps -ef | grep java
```

The command should return something similar to the following:

```
.../jre/bin/java -XX:MaxPermSize=128m –Xms1024M -Xmx2048M –server …
```

This may be useful only if you are running 1 SA, as multiple SAs will wait for memory to become available before launching.

4.10 *Using the Hindcast Robot*

The hindcast generation process can be automated through the use of the Hindcasting Robot. The steps are detailed in the *Hindcast Robot Manual*. In summary, they are:

- Install the robot software.
- Add the location of the robot (fews_hindcast_robot.jar) and the robot libraries (sikuli_libs) to your PATH.
- Edit a config file (config.txt below) to tell the robot the Start T0/End T0 you want to run (this can span several years). Also identify where your Export (output) directory is, as the robot checks for, and re-runs, skipped T0s.
- Since the robot controls the mouse, it is recommended that it be run inside an NXClient to give the robot a virtual desktop in which to execute.
- Start the robot and wait for it to complete. The command is similar to the following:

```
../jre/bin/java -jar fews_hindcast_robot.jar ./config.txt
```

A default forecastLength can also be set (but it is not required):

```
cd <test_root>/Config/DisplayConfigFiles/
nedit ManualForecastDisplay.xml
```

Add the following lines to the bottom of the file, before the closing XML element </manualForecastDisplay>:

```
<forecastLength unit="day" multiplier="330"/>
```

If multiple robots are to be run on a single machine, it is recommended that the number of hindcasting SAs used be at most one-half of the available CPUs on the machine. Each SA needs system CPU time to write /tmp directories and allocate Java threads. The system load (uptime) increased by 2 for every SA run during experimentation, so 3 SAs = an additional load of 6. The higher load didn't seem to affect the T0 runtimes too much. The first SA took about 2 minutes per T0 = 12 hours per hindcast year. The last SA took about 3 minutes per T0 = 18 hours per hindcast year.

4.11 *Using the FEWS Templates for Time Series Exporting*

If there are many locations for which files are to be exported and the exports are almost always identical, then a template approach may work.

Assume the hindcast workflow calls 3 export workflows. For example:

```
<activity>
  <runIndependent>false</runIndependent>
  <workflowId>MEFP_Export</workflowId>
</activity>
<activity>
  <runIndependent>false</runIndependent>
  <workflowId>HEFS_Export</workflowId>
</activity>
<activity>
  <runIndependent>false</runIndependent>
  <workflowId>EnsPost_Export</workflowId>
</activity>
```

This example will use four export templates with the following moduleInstanceIds:

MEFP_Precip_Export
MEFP_Temp_Export
HEFS_Streamflow_Export
EnsPost_Streamflow_Export

Assume there are 2 locations for which data is to be export:

DCWC1HOF
ORDC1LLF

The MEFP_Export workflow will call MEFP_Precip_Export once for each of the two locations, and then call MEFP_Temp_Export once for each of the two locations:

```
<activity>
   <properties>
      <string key="LOCATION_ID" value="DCWC1HOF"/>
   </properties>
   <runIndependent>false</runIndependent>
   <moduleInstanceId>MEFP_Precip_Export</moduleInstanceId>
</activity>
<activity>
   <properties>
      <string key="LOCATION_ID" value="ORDC1LLF"/>
   </properties>
```

```
    <runIndependent>false</runIndependent>
    <moduleInstanceId>MEFP_Precip_Export</moduleInstanceId>
  </activity>
  <activity>
    <properties>
      <string key="LOCATION_ID" value="DCWC1HOF"/>
    </properties>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>MEFP_Temp_Export</moduleInstanceId>
  </activity>
  <activity>
    <properties>
      <string key="LOCATION_ID" value="ORDC1LLF"/>
    </properties>
    <runIndependent>false</runIndependent>
    <moduleInstanceId>MEFP_Temp_Export</moduleInstanceId>
  </activity>
```

The templates will then make use of the LOCATION_ID property defined in the properties XML element. The examples are below (note the use of $LOCATION_ID$):

EXAMPLE: MEFP_Precip_Export.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>MEFP precip Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/mefp</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
<!--
            <exportFile>%TIME0%_$LOCATION_ID$_precip.fi</exportFile>
            <exportBinFile>true</exportBinFile>
-->
            <exportFile>%TIME0%_$LOCATION_ID$_precip.xml</exportFile>
            <exportBinFile>false</exportBinFile>

            <timeSeriesSets>

               <timeSeriesSet>
                  <moduleInstanceId>FeatherYuba_MEFP_FMAP_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>FMAP</parameterId>
                  <locationId>$LOCATION_ID$</locationId>
                  <timeSeriesType>external forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="6" startOverrulable="false" end="360"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start='1950' end='2009'/>
               </timeSeriesSet>

            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

EXAMPLE: MEFP_Temp_Export.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>MEFP temp Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/mefp</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
<!--
            <exportFile>%TIME0%_$LOCATION_ID$_temp.fi</exportFile>
            <exportBinFile>true</exportBinFile>
-->
            <exportFile>%TIME0%_$LOCATION_ID$_temp.xml</exportFile>
            <exportBinFile>false</exportBinFile>

            <timeSeriesSets>

               <timeSeriesSet>
                  <moduleInstanceId>MEFP_FMAT_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>FMAT</parameterId>
                  <locationId>$LOCATION_ID$</locationId>
                  <timeSeriesType>external forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="6" startOverrulable="false" end="360"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start='1950' end='2009'/>
               </timeSeriesSet>

            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

The HEFS_Streamflow_Export module is a little different because the moduleInstanceId changes with the locationID. In HEFS_Streamflow_Export module, each activity passes two properties:

```
<activity>
  <properties>
    <string key="LOCATION_ID"     value="DCWC1"/>
    <string key="MOD_INSTANCE_ID" value="UNITHG_DCWC1H_DCWC1_Forecast"/>
  </properties>
  <runIndependent>false</runIndependent>
  <moduleInstanceId>HEFS_Streamflow_Export</moduleInstanceId>
</activity>
<activity>
  <properties>
    <string key="LOCATION_ID"     value="ORDC1"/>
    <string key="MOD_INSTANCE_ID" value="ADDSUB_ORDC1L_AS1_Forecast"/>
  </properties>
  <runIndependent>false</runIndependent>
  <moduleInstanceId>HEFS_Streamflow_Export</moduleInstanceId>
</activity>
```

The templates will then make use of the LOCATION_ID and MOD_INSTANCE_ID properties defined in the properties XML element. The example is below (note the use of $LOCATION_ID$ and $MOD_INSTANCE_ID$):

EXAMPLE: HEFS_Streamflow_Export.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>Streamflow Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/streamflow</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
<!--
            <exportFile>%TIME0%_$LOCATION_ID$_streamflow.fi</exportFile>
            <exportBinFile>true</exportBinFile>
-->
            <exportFile>%TIME0%_$LOCATION_ID$_streamflow.xml</exportFile>
            <exportBinFile>false</exportBinFile>

            <timeSeriesSets>

               <timeSeriesSet>
                  <moduleInstanceId>$MOD_INSTANCE_ID$</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>SQIN</parameterId>
                  <locationId>$LOCATION_ID$</locationId>
                  <timeSeriesType>simulated forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="6" startOverrulable="false" end="360"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>MEFP</ensembleId>
                  <ensembleMemberIndexRange start='1950' end='2009'/>
               </timeSeriesSet>

            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

Like the MEFP export modules, the EnsPost_Streamflow_Export module uses only a single property:

```
<activity>
   <properties>
      <string key="LOCATION_ID" value="DCWC1"/>
   </properties>
   <runIndependent>false</runIndependent>
   <moduleInstanceId>EnsPost_Streamflow_Export</moduleInstanceId>
</activity>
<activity>
   <properties>
      <string key="LOCATION_ID" value="ORDC1"/>
   </properties>
   <runIndependent>false</runIndependent>
```

```
    <moduleInstanceId>EnsPost_Streamflow_Export</moduleInstanceId>
  </activity>
```

The example is below (note the use of $LOCATION_ID$):

EXAMPLE: EnsPost_Streamflow_Export.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<generalAdapterRun xmlns="http://www.wldelft.nl/fews"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wldelft.nl/fews http://chps1/schemas/generalAdapterRun.xsd">
   <general>
      <description>EnsPost Export</description>
      <piVersion>1.5</piVersion>
      <rootDir>%REGION_HOME%/Export</rootDir>
      <workDir>%ROOT_DIR%/Export</workDir>
      <exportDir>%REGION_HOME%/Export/enspost</exportDir>
      <exportDataSetDir>REGION_HOME%/Export</exportDataSetDir>
      <exportIdMap>IdExportLAGK</exportIdMap>
      <importDir>%ROOT_DIR%/output</importDir>
      <importIdMap>IdImportLAGK</importIdMap>
      <dumpFileDir>$GA_DUMPFILEDIR$</dumpFileDir>
      <dumpDir>%ROOT_DIR%</dumpDir>
      <diagnosticFile>%ROOT_DIR%/output/diag.fi</diagnosticFile>
      <missVal>-999</missVal>
      <modelTimeStep unit="hour" multiplier="6"/>
   </general>
   <activities>
      <exportActivities>
         <exportTimeSeriesActivity>
<!--
            <exportFile>%TIME0%_$LOCATION_ID$_enspost.fi</exportFile>
            <exportBinFile>true</exportBinFile>
-->
            <exportFile>%TIME0%_$LOCATION_ID$_enspost.xml</exportFile>
            <exportBinFile>false</exportBinFile>

            <timeSeriesSets>

               <timeSeriesSet>
                  <moduleInstanceId>$LOCATION_ID$_ENSPOST_Forecast</moduleInstanceId>
                  <valueType>scalar</valueType>
                  <parameterId>SQIN</parameterId>
                  <locationId>$LOCATION_ID$</locationId>
                  <timeSeriesType>simulated forecasting</timeSeriesType>
                  <timeStep unit="hour" multiplier="6"/>
                  <relativeViewPeriod unit="hour" start="6" startOverrulable="false" end="360"
endOverrulable="false"/>
                  <readWriteMode>read only</readWriteMode>
                  <ensembleId>HEFSENSPOST</ensembleId>
                  <ensembleMemberIndexRange start='1950' end='2009'/>
               </timeSeriesSet>

            </timeSeriesSets>
         </exportTimeSeriesActivity>
      </exportActivities>

      <executeActivities>
      </executeActivities>

      <importActivities>
      </importActivities>

   </activities>
</generalAdapterRun>
```

The final step is to define the descriptors for the four export templates within the ModuleInstanceDescriptors.xml file as GeneralAdapter runs:

```
<moduleInstanceGroup id="Export_Templates">
   <moduleInstanceDescriptor id="MEFP_Precip_Export">
          <moduleId>GeneralAdapter</moduleId>
   </moduleInstanceDescriptor>
   <moduleInstanceDescriptor id="MEFP_Temp_Export">
          <moduleId>GeneralAdapter</moduleId>
   </moduleInstanceDescriptor>
   <moduleInstanceDescriptor id="HEFS_Streamflow_Export">
          <moduleId>GeneralAdapter</moduleId>
   </moduleInstanceDescriptor>
   <moduleInstanceDescriptor id="EnsPost_Streamflow_Export">
          <moduleId>GeneralAdapter</moduleId>
   </moduleInstanceDescriptor>
</moduleInstanceGroup>
```

4.12 *Miscellaneous Tips*

The following are miscellaneous tips that OHD has found useful.

4.12.1 Exporting Time Series to MEFP When Hindcasting

If you're not using a time series when generating a hindcast, comment it out in the exportTimeSeriesActivity of the module configuration files for executing the MEFPEnsembleGeneratorModelAdapter.

However, at least one timeSeriesSet must be specified in the exportTimeSeriesActivity for FEWS to execute the adapter, even if it will export all missing values when run for a *hindcast T0*.

4.12.2 Debugging an MEFP/EnsPost Problem

In the exportRunFileActivity for the module that is causing the problem, set the printDebugInfo property to a value greater than 0 to get DEBUG-level log messages:

```
<int key="printDebugInfo" value="1"/>
```

This will allow for more information to be displayed in the **Logs** panel and log.txt file when a module is executed in debug mode through the **Manual Forecast Dialog**.

Use the testing property to ensure that MEFP generates the same output when executed again in order to recreate the problem:

```
<string key="testing" value="1423569318720"/>
```

The value of the testing property defines the random number seed to use when executing MEFP. The seed that is used for any run of the MEFP is output to the log.txt file as an INFO level message. Look for a line in log.txt (or diag.xml when executed in debug mode) similar to the following:

```
Random number generator for thread 1 initialized with the seed 1423569318720
```

By setting the testing property to have a value equal to the seed (highlighted above), the MEFP will use the same sequence of random numbers when generating an ensemble and, therefore, generate the same output assuming all other inputs are identical.

4.12.3 Restrict the Workflow to Your Locations

If hindcasts are to be generated for only a subset of locations for which HEFS can be run, then comment-out the unused locations in the following files:

<*configuration_dir*>/RegionConfigFiles/…
LocationSets.xml
ModuleInstanceSets.xml
<*configuration_dir*>/ModuleConfigFiles/hefs/… (location may vary)
ForecastGroup_UpdateStates.xml: the update states workflow to be executed in Section 3.2.2.
HEFS_PreProcessing_Forecast.xml: The preprocessing workflow that executes the merge transformations mentioned in Section 2.3.1 and elsewhere.

This will speed up your hindcast by eliminating unnecessary processing.

4.12.4 Run on a Local Disk

If the *hindcasting SA* is placed on a local disk (relative to the machine on which it is executing), it will run about 30% faster due to time saved performing disk I/O. To identify the local disks, run the Linux command:

```
df -h
```

Look for file systems that start with "/dev". Move the SA to one of those file systems and replace all links with copies of what they link to. The links to replace are:

bin
jre
ohdPlugins
<*region_dir*>/patch.jar
<*region_dir*>/Models/ohd/bin

The tempDir specified in sa.global_properties, localDataStore, and your Export directory should already be local. Check that the file system has enough space to store your exports.

4.12.5 Example OHD Run Times (on a Local Disk)

Run property	Scenario 1	Scenario 2	Scenario 3
Forecast horizon (days)	15	15	365
RFC	MARFC	CNRFC	MARFC
Number of MEFP basins	14	28	14
Number of flow basins	14	15	14
Years (# years)	1988-1998 (10)	1985-1995 (10)	1988-1998 (10)
HEFS components	All (no GraphGen)	All (no GraphGen)	All (no GraphGen)
Frequency of T0s (days)	1	1	5
Model timestep (hours)	6	1	6
Forcing sources	GEFS	GEFS	GEFS+CFSv2+CLIM
Runtime per T0 (mins)	0.75	1.4	3.2
Runtime per year (mins)	278	517	240
Total run time (mins)	2780	5170	2400

MEFP as % of run time	32	60	-
localDataStore (GB)	16	38	41
PI-XML total export (GB)	43	71	14.5

4.12.6 Change Simulated Forecasting Time Series to Temporary

You can save localDataStore space and speed up your hindcast by changing all simulated forecasting time series, as defined by a timeSeriesType XML element of “simulated forecasting” within the timeSeriesSet elements defining input and output, to be temporary:

```
<timeSeriesType>temporary</timeSeriesType>
```

Temporary time series are not written to the localDataStore, they are kept in memory, instead, and discarded when the hindcast is complete. Sample savings for an 18 year hindcast, over a single location, are:

	localDataStore increase (GB)	Runtime (hrs:min)
Simulated forecasting	1.6	6:27
Temporary	0.056	4:33

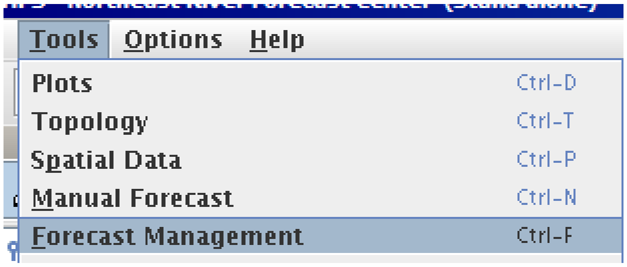
Additionally, it is recommended you execute a test hindcast in order to identify external forecasting time series (i.e., timeSeriesType XML element of “external forecasting”) that are being written to the localDataStore. In many cases, those can also be changed to ‘temporary’ for additional savings in time and disk space.

When making a change to a timeSeriesType element, all references to that time series must be changed. For example, if the output from a model is modified to be a temporary time series, then all modules that use that model output as input must be modified to be temporary as well.

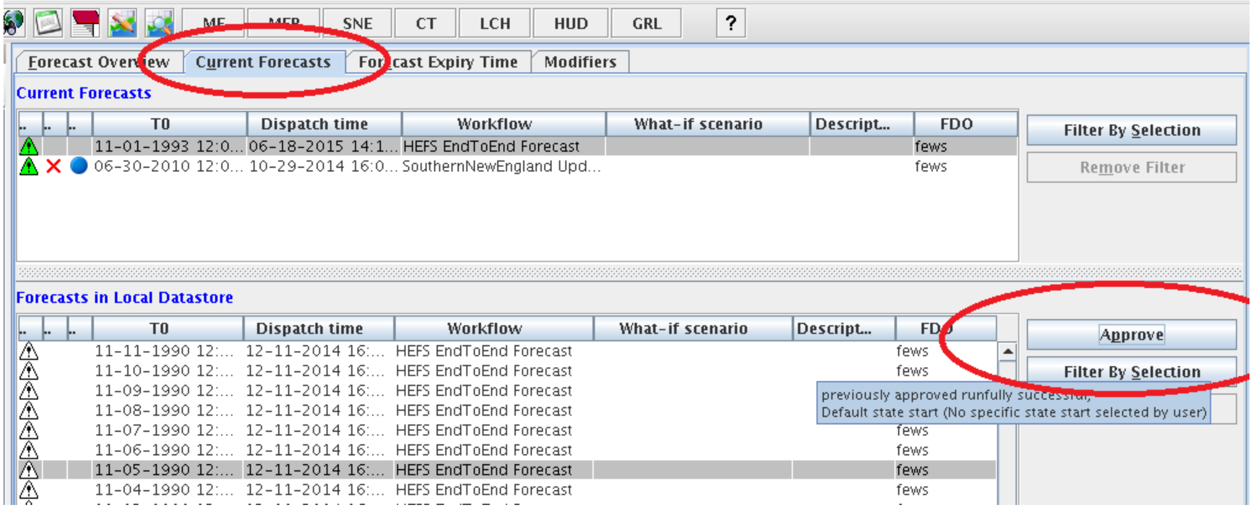
.

4.12.7 Exporting from a Previous Hindcast

To export from a hindcast stored in the localDataStore, make it the current forecast. In the **Tools Menu**, select **Forecast Management**:



Select the **Current Forecasts** tab. In the **Forecasts in Local Database Table**, select the forecast you wish to export (here 11-05-1990), and click the **Approve Button**:



Current Forecasts

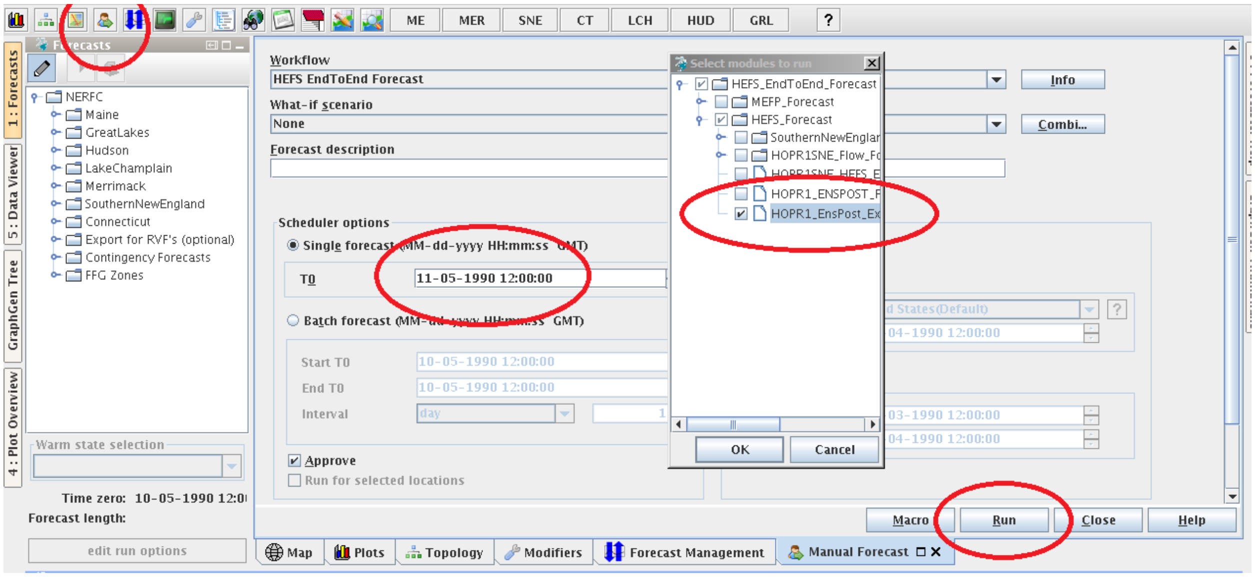
..	..	..	T0	Dispatch time	Workflow	What-if scenario	Descript...	FDO
⚠			11-01-1993 12:0...	06-18-2015 14:1...	HEFS EndToEnd Forecast			fews
⚠	✗	●	06-30-2010 12:0...	10-29-2014 16:0...	SouthernNewEngland Upd...			fews

Forecasts in Local Datastore

..	..	..	T0	Dispatch time	Workflow	What-if scenario	Descript...	FDO
⚠			11-11-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			fews
⚠			11-10-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			fews
⚠			11-09-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			
⚠			11-08-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			
⚠			11-07-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			fews
⚠			11-06-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			fews
⚠			11-05-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			fews
⚠			11-04-1990 12:...	12-11-2014 16:...	HEFS EndToEnd Forecast			fews

previously approved runfully successful,
Default state start (No specific state start selected by user)

Return to the **Manual Forecast Dialog**, set the forecast T0 to match the export date, and make it so that only the export modules are executed in the hindcast workflow. To do so, press F12 after selecting the workflow from the **Workflow Choicebox** and click **Select modules to run**. Uncheck all modules but the export workflow modules. In the example screenshot below, only HOPR1_EnsPost_Export is executed.



Workflow: HEFS EndToEnd Forecast

What-if scenario: None

Forecast description:

Scheduler options

● Single forecast (MM-dd-yyyy HH:mm:ss GMT)

T0 11-05-1990 12:00:00

○ Batch forecast (MM-dd-yyyy HH:mm:ss GMT)

Start T0 10-05-1990 12:00:00

End T0 10-05-1990 12:00:00

Interval day 1

☑ Approve

☐ Run for selected locations

Time zero: 10-05-1990 12:0

Forecast length:

Select modules to run

- ☑ HEFS_EndToEnd_Forecast
  - ☐ MEFP_Forecast
  - ☑ HEFS_Forecast
    - ☐ SouthernNewEnglar
    - ☐ HOPR1SNE_Flow_Fc
    - ☐ HOPR1SNE_HEFS_E
    - ☐ HOPR1_ENSPOST_F
    - ☑ HOPR1_EnsPost_Ex

4.12.8 Storing a Hindcast

When storing hindcast output, you should include the CHPS SA that generated it, so you save the parameters that created it. Storing the localDataStore is also recommended if you wish to re-run without recreating the warm states.

4.13 *Frequently Asked Questions*

1. **(EnsPost) If there is missing historical QME/QIN/QINE data, can we use that time period for hindcasting? Or can we only hindcast when there is a complete series of data, with no missing values?**

The MEFP ensembles can be generated with missing historical observed MAP and MAT, but the EnsPost will fail (with an appropriate error message) if the historical observed QME/QIN/QINE, and/or SQIN cannot be found.

2. **(MEFP) I am seeing the following. What does it mean?**

> Error executing model: Canonical event values for the CFSv2 forecast source and forecast time 1992-02-04 12:00:00 GMT were not found for the location AMEI4 (MAP)

MEFP is unable to extract required input data for the specified forecast source and the specified *hindcast T0*. First, ensure that the hindcast T0 falls on a day for which CFSv2 data is available; see Section 3.3.1 for a complete description of the CFSv2 data availability and the leap-year problem.

Second, check the MEFP parameter file. The CFSv2 forecast source must have been included when estimating the parameters. To check this, use the MEFPPE **Time Series Diagnostics Panel** to view the reforecast time series for the location. Then, load the parameters and view the **Parameter Diagnostics** to confirm that parameters were estimated for CFSv2. Alternatively, load the estimation options from the parameter file and check the **Estimation Options Panel** to see if the CFSv2 forecast source is checked.

Instructions are provided in the *MEFP User's Manual*.

3. **(EnsPost) How do I use USGS QIN instead of QME as input to the HEFSEnsPostModelAdapter?**

First, import the QIN into the localDataStore through any standard CHPS mechanism. Use the **Database Viewer** to verify that it imported correctly for the desired segment.

Next, identify the module to modify that executes the adapter. For each location included in the execution of the adapter, the following input must be provided **<u>in order</u>**:

- The observed streamflow time series from which the latest observation is extracted.

- The streamflow ensemble to post-process.

To use the USGS QIN as input, identify and modify the exportTimeSeriesActivity timeSeriesSet XML element that corresponds to the first input, above, so that it matches the imported QIN data.

4. (EnsPost) Our QME data is 24-hr data at 12Z. Can the ENSPOST adapter handle 12Z as a time step, or do I need to make it a 24-hr time step prior to running the adapter?

EnsPost will import the 24-hr data at a 12Z time step.